A symbolic approach to the self-triggered design for networked control systems

Kazumune Hashimoto, Adnane Saoud, Masako Kishida, Toshimitsu Ushio and Dimos V. Dimarogonas

Abstract—**In this paper, we investigate novel self-triggered controllers for nonlinear control systems with reachability and safety specifications. To synthesize the self-triggered controller, we leverage the notion of symbolic models, or abstractions, which represent abstracted expressions of control systems. The symbolic models will be constructed through the concepts of approximate alternating simulation relations, based on which, and by employing a reachability game, the self-triggered controller is synthesized. We illustrate the effectiveness of the proposed approach through numerical simulations.**

Index Terms—**Self-triggered control, reachability and safety, symbolic models.**

I. Introduction

EVENT and self-triggered control have been prevalent in recent years as the useful control strategies to reduce communication resources for networked control systems (NCSs) [1]. The key idea of these approaches is that, network transmissions from sensors to the remote controller are given based on some criteria, such as stability or some control performances. Introducing the event and self-triggered control has been proven effective, since it leads to the potential energy-savings of battery powered devices by mitigating the communication load for NCSs.

So far, a wide variety of event and self-triggered controllers has been provided from theory to practical implementations, see, e.g., [2], [3] for survey papers. In this paper, we are particularly interested in designing a self-triggered strategy under *reachability* and *safety* specifications. In other words, our goal is to design a self-triggered controller, such that the state trajectory enters a target set in finite time (reachability), while at the same time remaining inside a safety set for all times (safety). To the best of our knowledge, event and self-triggered strategies that can accommodate reachability and safety specifications have been provided only in a few works, see, e.g., [4]–[9]. Event/self-triggered controllers based on reachability analysis or controlled invariant sets have been proposed in [4]–[8]. However, a fundamental assumption required in these previous approaches is that the safety set is *convex*; in some practical applications, such as robot motion planning, safety sets are typically *non-convex* due to the presence of obstacles. Hence, the above previous approaches may be limited for certain practical applications. A self-triggered algorithm that deals with non-convex safety sets has been proposed in [9]. In this previous work, a sufficient condition to generate a feasible communication scheduling has been derived, based on the assumption that there exists an *δ-ISS Lyapunov function* for the control system. However, assuming the existence of such a Lyapunov function limits the class of control systems, since it must ensure contractive behaviors between any pair of state trajectories. Therefore, designing an event/self-triggered controller that can accommodate non-convex safety constraints and that does not require any stability assumptions is still a challenging problem, which is our main objective and is tackled in this paper.

In this paper, we investigate a new self-triggered controller that takes different approaches from the previous works in the literature. The main contribution is to employ the notion of *symbolic models* (see, e.g., [10]). Roughly speaking, the symbolic model represents an abstracted expression of the control system, where each state of the symbolic model corresponds to an aggregate of states of the control system. The utilization of symbolic models is motivated by the fact that the self-triggered controller for reachability and safety specifications can be synthesized by employing algorithmic techniques from supervisory control, such as reachability/safety games, which, in particular, allow to deal with non-convex safety sets. More importantly, the only assumption required for the controller synthesis is Lipschitz continuity, and it does not require any stability assumption such as the one considered in [9]. Hence, the proposed approach is advantageous over the aforementioned previous works, in the sense that it deals with the non-convexity of safety sets and can be applied to a wide class of (Lipschitz) control systems.

Our approach is also related to several abstraction schemes, see, e.g., [11]–[18]; in particular, it may be closely related to [11], [12], in which some methods of constructing symbolic models with event-triggered strategies have been provided. Note that our approach differs from these previous results in the following sense. In the previous results, for example in [11], the authors provided a way to construct symbolic models with a *given* event-triggered strategy. On the other hand, our approach aims at *synthesizing* a self-triggered controller through the construction of symbolic models, such that the

Kazumune Hashimoto and Toshimitsu Ushio are with the Graduate School of Engineering Science, Osaka University, Osaka, Japan (e-mail: kazumune@kth.se, ushio@sys.es.osaka-u.ac.jp)

Adnane Saoud is with Laboratoire des Signaux et Systèmes and with Laboratoire Spécification Vérification, CNRS, Paris-Saclay, France (e-mail: adnane.saoud@l2s.centralesupelec.fr)

Masako Kishida is with National Institute of Informatics (NII), Tokyo, Japan (e-mail: kishida@nii.ac.jp)

Dimos V. Dimarogonas is with the School of Electrical Engineering, KTH Royal Institute of Technology, Sweden (e-mail: dimos@kth.se)

The authors are supported by ERATO HASUO Metamathematics for Systems Design Project (No. JPMJER1603), JST, Labex DigiCosme (project ANR-11-LABEX-0045-DIGICOSME) operated by ANR, the Swedish Research Council (VR), the Swedish Foundation for StrategicResearch (SSF), the H2020 ERC Starting Grant BUCOPHSYS and the Knut och Alice Wallenberg Foundation (KAW).

reachability and safety specifications are fulfilled. In addition to [11], [12], several approaches to obtain symbolic models for NCSs have been also proposed, see e.g., [16], [18]; however, none of these works provided a way to synthesize event/self-triggered controllers, which will be the main objective considered in this paper.

Notation. Let $\mathbb{N}$, $\mathbb{N}_{\geq 0}$, $\mathbb{N}_{>0}$, $\mathbb{N}_{a:b}$ be the set of integers, non-negative integers, positive integers, and the set of integers in the interval $[a,b]$, respectively. Let $\mathbb{R}$, $\mathbb{R}_{\geq 0}$, $\mathbb{R}_{>0}$ be the set of reals, non-negative reals and positive reals, respectively. We denote by $\|\cdot\|$ the Euclidean norm. For given $x \in \mathbb{R}^n$ and $\varepsilon \in \mathbb{R}_{\geq 0}$, let $\mathcal{B}_\varepsilon(x) \subset \mathbb{R}^n$ be the ball set given by $\mathcal{B}_\varepsilon(x) = \{x' \in \mathbb{R}^n \mid \|x' - x\| \leq \varepsilon\}$. For given $X \subseteq \mathbb{R}^n$ and $\eta > 0$, denote by $[X]_\eta \subset \mathbb{R}^n$ the lattice in X with the quantization parameter η, i.e., $[X]_\eta = \{x \in X \mid x_i = \frac{2\eta}{\sqrt{n}} a_i,\ a_i \in \mathbb{N},\ i = 1, 2, \ldots, n\}$, where $x_i \in \mathbb{R}$ is the i-th element of x. Given $X \subset \mathbb{R}^n$, let $\mathsf{Int}_\varepsilon(X) = \{x \in X \mid \mathcal{B}_\varepsilon(x) \subseteq X\}$, i.e., $\mathsf{Int}_\varepsilon(X)$ is the set of all states in X, such that these are ε-away from the boundary of X. Given $X \subseteq \mathbb{R}^n$, denote by X^* the set of all finite sequences of elements in X. Given $x \in \mathbb{R}^n$, $X \subseteq \mathbb{R}^n$, denote by $\mathsf{Nearest}_X(x)$ the closest points in X to x, i.e., $\mathsf{Nearest}_X(x) = \arg\min_{x' \in X} \|x - x'\|$. Given a set X, denote by 2^X the power set of X that represents the collection of all subsets of X.

II. Problem formulation

A. System description

Consider a networked control system shown in Fig. 1, where the plant and the controller are connected over a communication network. We assume that the dynamics of the plant is given by the following nonlinear discrete-time systems:

$$x_{k+1} = f(x_k, u_k), \quad x_0 \in X_0, \ u_k \in U \tag{1}$$

for all $k \in \mathbb{N}_{\geq 0}$, where $x_k \in \mathbb{R}^{n_x}$ is the state, $u_k \in \mathbb{R}^{n_u}$ is the control input, $X_0 \subset \mathbb{R}^{n_x}$ is the set of initial states, $U \subset \mathbb{R}^{n_u}$ is the set of control inputs, and $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ is the function that represents the underlying model of the plant. Throughout the paper, we assume that X_0 and U are both compact sets. Moreover, we assume that the function f satisfies the following Lipschitz continuity:

Assumption 1. *The function $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ is Lipschitz continuous in $x \in \mathbb{R}^{n_x}$, i.e., there exists $L_x \in \mathbb{R}_{\geq 0}$, such that $\|f(x_1, u) - f(x_2, u)\| \leq L_x \|x_1 - x_2\|$ for all $x_1, x_2 \in \mathbb{R}^{n_x}, u \in U$.* □

We say that the sequence $x_0, x_1, x_2, \ldots \in \mathbb{R}^{n_x}$ is a *trajectory* of the system (1), if $x_0 \in X_0$ and there exist $u_0, u_1, u_2, \ldots \in U$ such that $x_{k+1} = f(x_k, u_k)$, $\forall k \in \mathbb{N}_{\geq 0}$. For simplicity of presentation, we denote by $\phi(x, u, m) \in \mathbb{R}^{n_x}$ the state that is reached from $x \in \mathbb{R}^{n_x}$ with $u \in U$ applied *constantly* for m time steps, i.e., $x^+ = \phi(x, u, m)$ iff there exist $x_0, \ldots, x_m \in \mathbb{R}^{n_x}$, such that $x_0 = x$, $x_{k+1} = f(x_k, u)$, $\forall k \in \mathbb{N}_{0:m-1}$, and $x_m = x^+$. The following result is an immediate consequence from Assumption 1, which will be utilized later in this paper:

Lemma 1. *For every $x_1, x_2 \in \mathbb{R}^{n_x}$, $u \in U$, and $m \in \mathbb{N}_{>0}$, $\|\phi(x_1, u, m) - \phi(x_2, u, m)\| \leq L_x^m \|x_1 - x_2\|$.* □

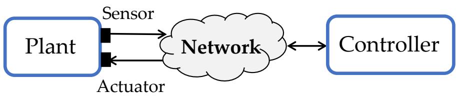

Fig. 1. Networked Control System.

B. Self-triggered strategy

Let us now provide the overview of the control strategies. First, let k_ℓ, $\ell \in \mathbb{N}_{\geq 0}$ with $k_0 = 0$, $k_{\ell+1} > k_\ell$, $\forall \ell \in \mathbb{N}_{\geq 0}$ be the communication time steps when the information is exchanged between the plant and the controller. In this paper, we employ a *self-triggered strategy* [1], which means that the controller is defined as a mapping from the state to the corresponding pairs of the control input and the inter-communication time step:

$$C : \mathbb{R}^{n_x} \to 2^{U \times \mathbb{N}_{>0}}. \tag{2}$$

That is, for each k_ℓ, $\ell \in \mathbb{N}_{\geq 0}$ the plant transmits the current state information x_{k_ℓ} to the controller, and the controller determines both the control input and the inter-communication time step as $\{u_{k_\ell}, m_\ell\} \in C(x_{k_\ell})$. Then, the controller transmits $\{u_{k_\ell}, m_\ell\}$ to the plant, and the plant applies u_{k_ℓ} until $k_{\ell+1} = k_\ell + m_\ell$, i.e., $u_k = u_{k_\ell}$, $\forall k \in \mathbb{N}_{k_\ell : k_{\ell+1}-1}$. Then, the next communication is given at $k_{\ell+1}$ and the same procedure as above is iterated. Given C, we say that the sequence $x_0, x_1, x_2, \ldots \in \mathbb{R}^{n_x}$ is a *controlled* trajectory of the system (1) induced by C, if $x_0 \in X_0$, $x_{k+1} = f(x_k, u_{k_\ell})$, $\forall k \in \mathbb{N}_{k_\ell : k_{\ell+1}-1}$, $\forall \ell \in \mathbb{N}_{\geq 0}$, where $k_0 = 0$, $k_{\ell+1} = k_\ell + m_\ell$, $\forall \ell \in \mathbb{N}_{\geq 0}$ and $\{u_{k_\ell}, m_\ell\} \in C(x_{k_\ell})$, $\forall \ell \in \mathbb{N}_{\geq 0}$.

C. Problem statement

Let $X_S \subset \mathbb{R}^{n_x}$ with $X_0 \subseteq X_S$ be the *safety set*, in which the trajectory must remain for all times. In addition, let $X_F \subset \mathbb{R}^{n_x}$ with $X_F \subseteq X_S$ be the *target set*, which the trajectory aims to reach in finite time. We assume that X_S, X_F are both compact and can be non-convex sets. Specifically, we illustrate the reachability and safety specifications by introducing the notion of *validity* of the controller C, which is defined below:

Definition 1. *Given (X_S, X_F), we say that the controller C is valid for the system* (1) *with the specification (X_S, X_F), if for every $x_0 \in X_0$ and the resulting controlled trajectory $x_0, x_1, x_2, \ldots,$ of* (1) *induced by C, the following statement holds: there exist N, $k_N \in \mathbb{N}_{\geq 0}$, such that*

(C1) $x_{k_N} \in X_F$;

(C2) $x_k \in X_S$, $\forall k \in \mathbb{N}_{0:k_N}$. □

That is, the controller C is valid if every initial state can be steered to the target set in finite time (C1), while at the same time remaining in the safety set for all times (C2). We now state the main problem to be solved in this paper:

Problem 1. *Given (X_S, X_F), synthesize a valid controller for the system* (1) *with the specification (X_S, X_F).* □

III. Constructing symbolic models

As the first step to solve Problem 1, this section presents a framework to obtain a symbolic model, which represents an abstracted expression of the control system.

A. Induced transition systems

Let us first define the notion of *transition systems*, which captures the control system described in (1):

Definition 2. *A transition system induced by the system* (1) *is a tuple* $\Sigma = (X, X_0, U, G, O)$*, where:*

- $X = \mathbb{R}^{n_x}$ *is a set of states;*
- $X_0 \subset X$ *is a set of initial states;*
- $U \subset \mathbb{R}^{n_u}$ *is a set of inputs;*
- $G : X \times U \to 2^X$ *is a transition map, where* $x^+ \in G(x, u)$ *iff* $x^+ = f(x, u)$*;*
- $O : X \times U \to 2^{X^*}$ *is an output map, where* $\{x^+\} \in O(x, u)$ *iff* $x^+ \in G(x, u)$*.* □

In Definition 2, the transition $x^+ \in G(x, u)$ means that the system evolves from x to x^+ by applying the control input u according to (1). The output map O is defined to produce, for each transition $x^+ \in G(x, u)$, the corresponding output, which is simply here given by $\{x^+\}$. Here, even though the system (1) is deterministic, the transition and the output maps are given by *set-valued* maps, in order to show that (1) can be captured within the general definition of transition systems.

Next, we extend Definition 2 by incorporating the self-triggered strategy. As stated in Section II-B, in the self-triggered strategy both the control input and the inter-communication time step are the decision variables that are determined by the controller, and the control input is applied constantly until the next communication time. In this paper, we express this fact by introducing the *augmented transition system* of Σ, which is formally defined below:

Definition 3. Let $\Sigma = (X, X_0, U, G, O)$ be a transition system defined in Definition 2. The *augmented transition system* of Σ is a tuple $\Sigma_A = (X, X_0, U, M, G_A, O_A)$, where:

- X is a set of states;
- X_0 is a set of initial states;
- U is a set of control inputs;
- $M = \mathbb{N}_{>0}$ is a set of inter-communication time steps;
- $G_A : X \times U \times M \to 2^X$ is a transition map, where $x^+ \in G_A(x, u, m)$ iff $x^+ = \phi(x, u, m)$;
- $O_A : X \times U \times M \to 2^{X^*}$ is an output map, where $\{x_1, \ldots, x_m\} \in O_A(x, u, m)$, iff $x_p \in G_A(x, u, p)$, $\forall p \in \mathbb{N}_{1:m}$. □

In essence, the augmented transition system Σ_A introduces the new transition map G_A and the output map O_A, which incorporate both the control input and the inter-communication time step as the decision variables. Specifically, $x^+ \in G_A(x, u, m)$ means that the system evolves from x to x^+ by applying $u \in U$ constantly for m time steps according to (1). In addition, $\{x_1, \ldots, x_m\} \in O_A(x, u, m)$ means that, while applying u constantly for m time steps from x, it produces the output $\{x_1, \ldots, x_m\}$ as the corresponding sequence of states generated according to the transition map G_A.

We say that $x_0, x_1, \ldots \in X$ is a *trajectory* of Σ_A, if $x_0 \in X_0$ and there exist $m_0, m_1, \ldots \in M$ and $u_{k_0}, u_{k_1}, \ldots \in U$, such that $\{x_{k_\ell+1}, \ldots, x_{k_{\ell+1}}\} \in O_A(x_{k_\ell}, u_{k_\ell}, m_\ell)$, $\forall \ell \in \mathbb{N}_{\geq 0}$, where $k_0 = 0$, $k_{\ell+1} = k_\ell + m_\ell$, $\forall \ell \in \mathbb{N}_{\geq 0}$. Given C, we say that a sequence $x_0, x_1, \ldots \in X$ is a *controlled* trajectory of Σ_A induced by C, if it is a trajectory of Σ_A with $\{u_{k_\ell}, m_\ell\} \in C(x_{k_\ell})$, $\forall \ell \in \mathbb{N}_{\geq 0}$. Note that the system (1) and Σ_A produce the same controlled trajectories, i.e., if $x_0, x_1, \ldots \in X$ is a controlled trajectory of the system (1) induced by C, so is of Σ_A induced by C, and vice versa.

B. Constructing symbolic models

Having defined the augmented transition system, we now construct a symbolic model of Σ_A. The symbolic model will be denoted as $\widetilde{\Sigma}^\varepsilon_A = (\widetilde{X}, \widetilde{X}_0, \widetilde{U}, \widetilde{M}, \widetilde{G}_A, \widetilde{O}_A)$, where $\widetilde{X}$ is a set of states, $\widetilde{X}_0$ is a set of initial states, $\widetilde{U}$ is a set of inputs, $\widetilde{M}$ is a set of inter-communication time steps, $\widetilde{G}_A$ is a transition map, and $\widetilde{O}_A$ is an output map. A more formal definition of $\widetilde{\Sigma}^\varepsilon_A$ will be given later in this section. To derive the symbolic model, we make use of the notion of *strong approximate alternating simulation relation* [16]:

Definition 4 (Strong ε-ASR)**.** Let $\Sigma_A = (X, X_0, U, M, G_A, O_A)$ and $\widetilde{\Sigma}^\varepsilon_A = (\widetilde{X}, \widetilde{X}_0, \widetilde{U}, \widetilde{M}, \widetilde{G}_A, \widetilde{O}_A)$ be two transition systems. A relation $R(\varepsilon) \subseteq \widetilde{X} \times X$ with $\varepsilon \in \mathbb{R}_{\geq 0}$ is called a *strong ε-approximate Alternating Simulation Relation* (strong ε-ASR for short) from $\widetilde{\Sigma}^\varepsilon_A$ to Σ_A, if the following conditions hold:

(D1) For every $\tilde{x}_0 \in \widetilde{X}_0$, there exists $x_0 \in X_0$ such that $(\tilde{x}_0, x_0) \in R(\varepsilon)$;

(D2) For every $(\tilde{x}, x) \in R(\varepsilon)$, we have $\|\tilde{x} - x\| \leq \varepsilon$;

(D3) For every $(\tilde{x}, x) \in R(\varepsilon)$ and for every $\tilde{u} \in \widetilde{U}$, $\widetilde{m} \in \widetilde{M}$, there exist $u = \tilde{u} \in U$, $m = \widetilde{m} \in M$, such that the following holds: $\{x_1, \ldots, x_m\} \in O_A(x, u, m)$ implies the existence of $\{\tilde{x}_1, \ldots, \tilde{x}_{\widetilde{m}}\} \in \widetilde{O}_A(\tilde{x}, \tilde{u}, \widetilde{m})$, such that $(\tilde{x}_p, x_p) \in R(\varepsilon)$, $\forall p \in \mathbb{N}_{1:m} (= \mathbb{N}_{1:\widetilde{m}})$. □

The notion of *strong* ε-ASR is strong with respect to the standard notion of ε-ASR [15], in the sense that it must ensure the existence of the same control input $u = \tilde{u}$ (and $m = \widetilde{m}$) for the similarity condition of (D3). The concept of (strong) ε-ASR is useful for the controller synthesis in the following sense. Suppose that $\widetilde{\Sigma}^\varepsilon_A$ is constructed to guarantee the existence of a (strong) ε-ASR from $\widetilde{\Sigma}^\varepsilon_A$ to Σ_A. Then, it is shown that, the existence of a valid controller for $\widetilde{\Sigma}^\varepsilon_A$ implies the existence of a valid controller for Σ_A. In other words, a controller for the symbolic model that satisfies reachability and safety specifications, can be *refined* to a controller for Σ_A and the system (1) that guarantees the same specifications (for a detailed discussion, see Section IV).

In the following, we provide an approach to construct $\widetilde{\Sigma}^\varepsilon_A$ as the abstraction of Σ_A. Given $\varepsilon \in \mathbb{R}_{\geq 0}$, the symbolic model is constructed to guarantee the existence of a strong ε-ASR from $\widetilde{\Sigma}^\varepsilon_A$ to Σ_A. First, let $\widetilde{X}_0 \subseteq X_0$, $\widetilde{X} \subseteq X$, $\widetilde{U} \subseteq U$ be given by $\widetilde{X}_0 = [X_0]_{\eta_x}, \widetilde{X} = [X]_{\eta_x}, \widetilde{U} = [U]_{\eta_u}$ for given $\eta_x, \eta_u \in \mathbb{R}_{>0}$. That is, we quantize the sets X_0, X, U with the quantization parameters η_x, η_u. Moreover, let $\widetilde{M} \subset M$ be given by $\widetilde{M} = \mathbb{N}_{1:M_{\max}}$ for a given $M_{\max} \in \mathbb{N}_{>0}$. That is, we restrict that the inter-communication time step cannot exceed $M_{\max}$. The above quantization as well as the restriction on the inter-communication time steps will ensure that the controller synthesis algorithm can be terminated with a finite number of iterations. Based on the above definitions, the symbolic model $\widetilde{\Sigma}^{\varepsilon}_{A}$ is formally defined as follows:

Definition 5. Let $\Sigma_A = (X, X_0, U, M, G_A, O_A)$ be the augmented transition system of Σ and let $\eta_x, \eta_u \in \mathbb{R}_{>0}$ and $M_{\max} \in \mathbb{N}_{>0}$ be the quantization parameters and the maximum inter-communication time step, respectively. For a given precision $\varepsilon \geq \eta_x$, a *symbolic model* of Σ_A is a tuple $\widetilde{\Sigma}^{\varepsilon}_{A} = (\widetilde{X}, \widetilde{X}_0, \widetilde{U}, \widetilde{M}, \widetilde{G}_A, \widetilde{O}_A)$, where

- $\widetilde{X}$ is a set of states;
- $\widetilde{X}_0$ is a set of initial states;
- $\widetilde{U}$ is a set of inputs;
- $\widetilde{M}$ is a set of inter-communication time steps;
- $\widetilde{G}_A : \widetilde{X} \times \widetilde{U} \times \widetilde{M} \to 2^{\widetilde{X}}$ is a transition map, where $\tilde{x}^+ \in \widetilde{G}_A(\tilde{x}, \tilde{u}, \tilde{m})$ iff $\tilde{x}^+ \in \mathcal{B}_{\varepsilon_{\tilde{m}}}(\phi(\tilde{x}, \tilde{u}, \tilde{m}))$ with $\varepsilon_{\tilde{m}} = L_x^{\tilde{m}} \varepsilon + \eta_x$;
- $\widetilde{O}_A : \widetilde{X} \times \widetilde{U} \times \widetilde{M} \to 2^{\widetilde{X}^*}$ is an output map, where $\{\tilde{x}_1, \ldots, \tilde{x}_{\tilde{m}}\} \in \widetilde{O}_A(\tilde{x}, \tilde{u}, \tilde{m})$ iff $\tilde{x}_p \in \widetilde{G}_A(\tilde{x}, \tilde{u}, p)$, $\forall p \in \mathbb{N}_{1:\tilde{m}}$. □

In contrast to Definition 3, the symbolic model deals with the states in $\widetilde{X}$ and control inputs in $\widetilde{U}$, and introduces the new transition and the output maps $\widetilde{G}_A$, $\widetilde{O}_A$. As shown in Definition 5, we have $\tilde{x}^+ \in \widetilde{G}_A(\tilde{x}, \tilde{u}, \tilde{m})$ iff $\tilde{x}^+ \in \mathcal{B}_{\varepsilon_{\tilde{m}}}(\phi(\tilde{x}, \tilde{u}, \tilde{m}))$. Intuitively, the set $\mathcal{B}_{\varepsilon_{\tilde{m}}}(\phi(\tilde{x}, \tilde{u}, \tilde{m}))$ represents the over-approximation of the reachable states from $\mathcal{B}_{\varepsilon}(\tilde{x})$ plus the quantization error η_x, by applying $\tilde{u}$ constantly for $\tilde{m}$ time steps. The output map $\widetilde{O}_A$ is defined to produce the set of sequence of states according to the transition map $\widetilde{G}_A$. The existence of a strong ε-ASR from $\widetilde{\Sigma}^{\varepsilon}_{A}$ to Σ_A, can be justified by the following result:

Lemma 2. *Let $\Sigma_A = (X, X_0, U, M, G_A, O_A)$ be the augmented transition system of Σ and let $\eta_x, \eta_u \in \mathbb{R}_{>0}$ and $M_{\max} \in \mathbb{N}_{>0}$ be the quantization parameters and the maximum inter-communication time step, respectively. Also, let $\widetilde{\Sigma}^{\varepsilon}_{A} = (\widetilde{X}, \widetilde{X}_0, \widetilde{U}, \widetilde{M}, \widetilde{G}_A, \widetilde{O}_A)$ be the symbolic model of Σ_A defined in Definition 5, where ε is the precision parameter with $\varepsilon \geq \eta_x$. Then, the relation*

$$R(\varepsilon) = \{(\tilde{x}, x) \in \widetilde{X} \times X \mid \|\tilde{x} - x\| \leq \varepsilon\} \tag{3}$$

is a strong ε-ASR from $\widetilde{\Sigma}^{\varepsilon}_{A}$ to Σ_A. □

Proof. Let $R(\varepsilon)$ be given by (3) with $\varepsilon \geq \eta_x$. Since $\widetilde{X}_0 \subseteq X_0$, for every $\tilde{x}_0 \in \widetilde{X}_0$ there exists $x_0 = \tilde{x}_0 \in X_0$ such that $\|\tilde{x}_0 - x_0\| = 0 \leq \varepsilon$. Hence, the condition (D1) in Definition 4 holds. The condition (D2) is satisfied from (3). To check (D3), consider any $(\tilde{x}, x) \in R(\varepsilon)$ and $(\tilde{u}, \tilde{m}) \in \widetilde{U} \times \widetilde{M}$. Let $u = \tilde{u} \in U$, $m = \tilde{m} \in M$ and consider $\{x_1, \ldots, x_m\} \in O_A(x, u, m)$. This implies from Definition 3 that $x_p = \phi(x, u, p)$, $\forall p \in \mathbb{N}_{1:m}$. Now, pick $\tilde{x}_p \in \mathsf{Nearest}_{\widetilde{X}}(x_p)$, $\forall p \in \mathbb{N}_{1:m}$. Since $\widetilde{X} = [X]_{\eta_x}$, it follows that $\|\tilde{x}_p - x_p\| \leq \eta_x$, $\forall p \in \mathbb{N}_{1:m}$, i.e., $(\tilde{x}_p, x_p) \in R(\varepsilon)$, $\forall p \in \mathbb{N}_{1:m} (= \mathbb{N}_{1:\tilde{m}})$. Then, we have

$$\begin{aligned} \|\tilde{x}_p - \phi(\tilde{x}, \tilde{u}, p)\| &\leq \|x_p - \phi(\tilde{x}, \tilde{u}, p)\| + \eta_x \\ &\leq L_x^p \|x - \tilde{x}\| + \eta_x \leq L_x^p \varepsilon + \eta_x, \end{aligned} \tag{4}$$

for all $p \in \mathbb{N}_{1:\tilde{m}}$, where we used $u = \tilde{u}$ and Lemma 1. Hence, $\tilde{x}_p \in \mathcal{B}_{\varepsilon_p}(\phi(\tilde{x}, \tilde{u}, p))$ with $\varepsilon_p = L_x^p \varepsilon + \eta_x$, $\forall p \in \mathbb{N}_{1:\tilde{m}}$ and thus $\tilde{x}_p \in \widetilde{G}_A(\tilde{x}, \tilde{u}, p)$, $\forall p \in \mathbb{N}_{1:\tilde{m}}$. Thus, $\{\tilde{x}_1, \ldots, \tilde{x}_{\tilde{m}}\} \in \widetilde{O}_A(\tilde{x}, \tilde{u}, \tilde{m})$ and $(\tilde{x}_p, x_p) \in R(\varepsilon)$, $\forall p \in \mathbb{N}_{1:m} (= \mathbb{N}_{1:\tilde{m}})$. Therefore, $R(\varepsilon)$ is a strong ε-ASR from $\widetilde{\Sigma}^{\varepsilon}_{A}$ to Σ_A. □

Algorithm 1: Controller synthesis for $\widetilde{\Sigma}^{\varepsilon}_{A}$.

```
Input  : $\widetilde{\Sigma}^{\varepsilon}_{A}$, $\widetilde{X}_S$, $\widetilde{X}_F$
Output: $\mathcal{L}(\tilde{x})$, $\forall \tilde{x} \in \widetilde{X}_S$, $\widetilde{P}_S \subseteq \widetilde{X}_S$
$n \leftarrow 0$, $P^{(n)} \leftarrow \widetilde{X}_F$;
$\mathcal{L}(\tilde{x}) \leftarrow 0$, $\forall \tilde{x} \in \widetilde{X}_F$;
while $P^{(n+1)} \neq P^{(n)}$ do
    $P^{(n+1)} \leftarrow P^{(n)} \cup \mathrm{Pre}(P^{(n)})$;
    $\mathcal{L}(\tilde{x}) \leftarrow n+1$, $\forall \tilde{x} \in P^{(n+1)} \backslash P^{(n)}$;
    $n \leftarrow n+1$;
end
$\widetilde{P}_S \leftarrow P^{(n)}$, $\mathcal{L}(\tilde{x}) \leftarrow \infty$, $\forall \tilde{x} \in \widetilde{X}_S \backslash \widetilde{P}_S$;
```

We say that $\tilde{x}_0, \tilde{x}_1, \ldots \in \widetilde{X}$ is a *trajectory* of $\widetilde{\Sigma}^{\varepsilon}_{A}$, if $\tilde{x}_0 \in \widetilde{X}_0$ and there exist $\tilde{m}_0, \tilde{m}_1, \ldots \in \widetilde{M}$ and $\tilde{u}_{k_0}, \tilde{u}_{k_1}, \ldots \in \widetilde{U}$, such that $\{\tilde{x}_{k_\ell + 1}, \ldots, \tilde{x}_{k_{\ell+1}}\} \in \widetilde{O}_A(\tilde{x}_{k_\ell}, \tilde{u}_{k_\ell}, \tilde{m}_\ell)$, $\forall \ell \in \mathbb{N}_{\geq 0}$, where $k_0 = 0$, $k_{\ell+1} = k_\ell + \tilde{m}_\ell$, $\forall \ell \in \mathbb{N}_{\geq 0}$.

IV. Self-triggered controller synthesis

In this section, we present a concrete algorithm to synthesize a valid controller for the system (1) as a solution to Problem 1. Given Σ_A, suppose that $\widetilde{\Sigma}^{\varepsilon}_{A}$ is constructed such that $R(\varepsilon)$ is a strong ε-ASR from $\widetilde{\Sigma}^{\varepsilon}_{A}$ to Σ_A, where $R(\varepsilon)$ is defined in (3). In what follows, we first synthesize a controller for $\widetilde{\Sigma}^{\varepsilon}_{A}$. Then, we synthesize a controller for Σ_A as well as for the system (1), based on the controller for $\widetilde{\Sigma}^{\varepsilon}_{A}$.

As with (2), let a controller for $\widetilde{\Sigma}^{\varepsilon}_{A}$ be given by $\widetilde{C} : \widetilde{X} \to 2^{\widetilde{U} \times \widetilde{M}}$. Given $\widetilde{C}$, we say that the sequence $\tilde{x}_0, \tilde{x}_1, \ldots \in \widetilde{X}$ is a *controlled* trajectory of $\widetilde{\Sigma}^{\varepsilon}_{A}$ induced by $\widetilde{C}$, if it is a trajectory of $\widetilde{\Sigma}^{\varepsilon}_{A}$ with $\{\tilde{u}_{k_\ell}, \tilde{m}_\ell\} \in \widetilde{C}(\tilde{x}_{k_\ell})$, $\forall \ell \in \mathbb{N}_{\geq 0}$. Moreover, let $\widetilde{X}_S = [\mathsf{Int}_{\varepsilon}(X_S)]_{\eta_x}$ and $\widetilde{X}_F = [\mathsf{Int}_{\varepsilon}(X_F)]_{\eta_x}$. This implies that

$$\tilde{x} \in \widetilde{X}_S (\text{or } \widetilde{X}_F), (\tilde{x}, x) \in R(\varepsilon) \Longrightarrow x \in X_S (\text{or } X_F). \tag{5}$$

Note that the sets $\widetilde{X}_S$, $\widetilde{X}_F$ are both finite, since X_S and X_F are both compact. The validity of the controller $\widetilde{C}$ for $\widetilde{\Sigma}^{\varepsilon}_{A}$ with $(\widetilde{X}_S, \widetilde{X}_F)$ is defined in the same way as Definition 1.

Now, consider the problem of synthesizing a valid controller for $\widetilde{\Sigma}^{\varepsilon}_{A}$ with the specification $(\widetilde{X}_S, \widetilde{X}_F)$. The controller for $\widetilde{\Sigma}^{\varepsilon}_{A}$ can be found by employing a reachability game [10], which is summarized in Algorithm 1. In the algorithm, the map $\mathrm{Pre} : 2^{\widetilde{X}_S} \to 2^{\widetilde{X}_S}$ is given by

$$\begin{aligned} &\mathrm{Pre}(P) = \{\tilde{x} \in \widetilde{X}_S \mid \exists (\tilde{u}, \tilde{m}) \in \widetilde{U} \times \widetilde{M} : \\ &\quad \forall \tilde{x}^+ \in \widetilde{G}_A(\tilde{x}, \tilde{u}, \tilde{m}),\ \tilde{x}^+ \in P,\ \widetilde{O}_A(\tilde{x}, \tilde{u}, \tilde{m}) \subseteq \widetilde{X}_S^*\}. \end{aligned} \tag{6}$$

Roughly speaking, $\mathrm{Pre}(P)$ is the set of all states in $\widetilde{X}_S$, for which there exists a pair of control input and inter-communication time step, such that all the corresponding successors according to the transition map $\widetilde{G}_A$ are inside P (i.e., $\tilde{x}^+ \in P$), while all the corresponding outputs are inside $\widetilde{X}_S^*$. As shown in the algorithm, we iteratively compute $P^{(n)}$, $n = 0, 1, \ldots$ until it converges to a fixed point set denoted as $\widetilde{P}_S$. Intuitively, $P^{(n)}$ is the set of all states in $\widetilde{X}_S$ that can reach

$\widetilde{X}_F$ within n transitions. The map $\mathcal{L} : \widetilde{X}_S \to \widetilde{M}$ is used to stack, for each state in $\widetilde{X}_S$, the number of transitions required to reach $\widetilde{X}_F$. Algorithm 1 is guaranteed to terminate with a finite number of iterations, since $\widetilde{X}_S$, $\widetilde{X}_F$, $\widetilde{U}$ and $\widetilde{M}$ are all finite. Based on Algorithm 1, we construct the controller for $\widetilde{\Sigma}^{\varepsilon}_A$ as follows:

$$\widetilde{C}(\tilde{x}) = \{(\tilde{u},\widetilde{m}) \in \widetilde{U}\times\widetilde{M} \mid \forall \tilde{x}^+ \in \widetilde{G}_A(\tilde{x},\tilde{u},\widetilde{m}) : \mathcal{L}(\tilde{x}^+) < \mathcal{L}(\tilde{x}),\ \widetilde{O}_A(\tilde{x},\tilde{u},\widetilde{m}) \subseteq \widetilde{X}^*_S\}, \qquad (7)$$

for all $\tilde{x} \in \widetilde{P}_S$. The following result shows that the above controller is proven valid with a certain initial condition:

Lemma 3. *Let $\widetilde{C}$ be the controller for $\widetilde{\Sigma}^{\varepsilon}_A$ as derived in* (7). *Then, $\widetilde{C}$ is valid for $\widetilde{\Sigma}^{\varepsilon}_A$ with the specification $(\widetilde{X}_S,\widetilde{X}_F)$, if $\widetilde{X}_0 \subseteq \widetilde{P}_S$.* □

Proof. Suppose that $\widetilde{X}_0 \subseteq \widetilde{P}_S$ and let $\tilde{x}_{k_0} \in \widetilde{X}_0$ with $k_0 = 0$ be the initial state of $\widetilde{\Sigma}^{\varepsilon}_A$. Since $\tilde{x}_{k_0} \in \widetilde{P}_S$, there exists $N \in \mathbb{N}_{\geq 0}$ such that $\mathcal{L}(\tilde{x}_{k_0}) = N$. This means that $\tilde{x}_{k_0} \in P^{(N)}\backslash P^{(N-1)} \subseteq \mathrm{Pre}(P^{(N-1)})$, and, from (6), there exists $(\tilde{u}',\widetilde{m}') \in \widetilde{U}\times\widetilde{M}$ such that $\forall \tilde{x}^+ \in \widetilde{G}_A(\tilde{x}_{k_0},\tilde{u}',\widetilde{m}')$, $\tilde{x}^+ \in P^{(N-1)}$ (i.e., $\mathcal{L}(\tilde{x}^+) \leq N-1$), $\widetilde{O}_A(\tilde{x}_{k_0},\tilde{u},\widetilde{m}) \subseteq \widetilde{X}^*_S$. Hence, the controller in (7) is feasible, i.e., $\widetilde{C}(\tilde{x}_{k_0}) \neq \emptyset$. Thus, by applying $\{\tilde{u}_{k_0},\widetilde{m}_0\} \in \widetilde{C}(\tilde{x}_{k_0})$, we obtain $\tilde{x}_{k_1} \in P^{(N-1)}$ and $\tilde{x}_k \in \widetilde{X}_S, \forall k \in \mathbb{N}_{k_0+1:k_1}$, where $k_1 = k_0 + \widetilde{m}_0$ and $\{\tilde{x}_{k_0+1},\ldots\tilde{x}_{k_1}\} \in \widetilde{O}_A(\tilde{x}_{k_0},\tilde{u}_{k_0},\widetilde{m}_0)$. In summary, $\tilde{x}_{k_0} \in P^{(N)} \implies \tilde{x}_{k_1} \in P^{(N-1)}$ and $\tilde{x}_k \in \widetilde{X}_S$, $\forall k \in \mathbb{N}_{k_0:k_1}$. By recursively applying the above for $k_0, k_1, \ldots$, we obtain $\tilde{x}_{k_0} \in P^{(N)} \implies \tilde{x}_{k_1} \in P^{(N-1)} \implies \cdots \implies \tilde{x}_{k_N} \in P^{(0)} = \widetilde{X}_F$ and $\tilde{x}_k \in \widetilde{X}_S$, $\forall k \in \mathbb{N}_{k_0:k_N}$. That is, the trajectory achieves reachability and safety. Thus, the controller in (7) is valid if $\widetilde{X}_0 \subseteq \widetilde{P}_S$. □

Now, we refine the controller for Σ_A as well as for the system (1), based on the controller as derived in (7):

$$C(x) = \left\{\widetilde{C}(\tilde{x}) \mid \tilde{x} \in \mathsf{Nearest}_{\widetilde{P}_S}(x)\right\},\ \forall x \in P_S, \qquad (8)$$

where $P_S = \{x \in X_S \mid \exists \tilde{x} \in \widetilde{P}_S,\ (\tilde{x},x) \in R(\varepsilon)\}$. The controller in (8) means that, for each $x \in P_S$, we pick the closest points in $\widetilde{P}_S$ to x, and associate the corresponding pairs of the control input and the inter-communication time step. Note that we have $P_S \subseteq X_S$ since $\widetilde{P}_S \subseteq \widetilde{X}_S$ and $\widetilde{X}_S = [\mathsf{Int}_{\varepsilon}(X_S)]_{\eta_x}$. Similarly to Lemma 3, the following result shows that the controller for the system (1) is proven valid if a certain initial condition is satisfied.

Theorem 1. *Let C be the controller as in* (8). *Then, C is valid for Σ_A as well as for the system* (1) *with the specification (X_S, X_F), if $X_0 \subseteq P_S$.* □

Proof. Suppose that $X_0 \subseteq P_S$ and let $x_{k_0} \in X_0$ with $k_0 = 0$ be the initial state of the system (1). Pick $\tilde{x}_{k_0} \in \mathsf{Nearest}_{\widetilde{P}_S}(x_{k_0})$. Since $x_{k_0} \in P_S$, it follows that $(\tilde{x}_{k_0}, x_{k_0}) \in R(\varepsilon)$. Moreover, since $\tilde{x}_{k_0} \in \widetilde{P}_S$ there exists an $N \in \mathbb{N}_{\geq 0}$ such that $\mathcal{L}(\tilde{x}_{k_0}) = N$. Now, pick $\{\tilde{u}_{k_0},\widetilde{m}_0\} \in \widetilde{C}(\tilde{x}_{k_0})$ and $\{u_{k_0}, m_0\} \in C(x_{k_0})$ with $u_{k_0} = \tilde{u}_{k_0}$, $m_0 = \widetilde{m}_0$. Let $x_{k_0+1},\ldots,x_{k_1}$ with $k_1 = k_0 + m_0$ be the controlled trajectory of Σ_A by applying u_{k_0} for m_0 steps, i.e., $\{x_{k_0+1},\ldots,x_{k_1}\} \in O_A(x_{k_0}, u_{k_0}, m_0)$. Then, from the fact that $R(\varepsilon)$ is a strong ε-ASR from $\widetilde{\Sigma}^{\varepsilon}_A$ to Σ_A and from Definition 4, there exists $\{\tilde{x}_{k_0+1},\ldots,\tilde{x}_{k_1}\} \in \widetilde{O}_A(\tilde{x}_{k_0},\tilde{u}_{k_0},\widetilde{m}_0)$, such that $(\tilde{x}_k, x_k) \in R(\varepsilon), \forall k \in \mathbb{N}_{k_0+1:k_1}$. Moreover, from the proof of Lemma 3, it follows that $\tilde{x}_{k_1} \in P^{(N-1)} \subseteq \widetilde{P}_S$ and $\tilde{x}_k \in \widetilde{X}_S$, $\forall k \in \mathbb{N}_{k_0:k_1}$. In summary, we obtain $(\tilde{x}_k, x_k) \in R(\varepsilon), \tilde{x}_k \in \widetilde{X}_S$, $\forall k \in \mathbb{N}_{k_0:k_1}$, and $\tilde{x}_{k_1} \in P^{(N-1)}$. Thus, by recursively applying the above procedure for $k_0, k_1, \ldots$, it follows that $(\tilde{x}_k, x_k) \in R(\varepsilon)$, $\tilde{x}_k \in \widetilde{X}_S$, $\forall k \in \mathbb{N}_{k_0:k_N}$ and $\tilde{x}_{k_N} \in P^{(0)} = \widetilde{X}_F$. Since $(\tilde{x}_k, x_k) \in R(\varepsilon), \forall k \in \mathbb{N}_{k_0:k_N}$ and by using (5), it then follows that $x_k \in X_S$, $\forall k \in \mathbb{N}_{k_0:k_N}$ and $x_{k_N} \in X_F$. Hence, the controller in (8) is valid for Σ_A if $X_0 \subseteq P_S$. Moreover, since Σ_A and the system (1) produce the same controlled trajectories (see Section III-A), the controller in (8) is also valid for the system (1) if $X_0 \subseteq P_S$. □

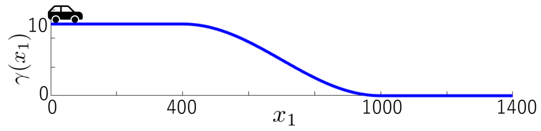

Fig. 2. Geometry of the road based on the road grade profile α.

TABLE I
PARAMETER SETTINGS

$\Delta : 1.0$	m_v: 1000	$O : [400, 1000] \times [12, 18]$
g: 9.8	$U : [-500, 500]$	$X_0 : [0, 50] \times [0, 18]$
d_1: 0.01	$X_S : X_P \backslash O$	$X_F : [1300, 1400] \times [0, 5]$
d_2: 0.1	$X_P : [0, 1400] \times [0, 18]$	

Remark 1 (On the selection of η_x, η_u)**.** *As η_x, η_u are selected smaller, the symbolic $\widetilde{\Sigma}^{\varepsilon}_A$ will be more precise with respect to Σ_A. Intuitively, this implies that the condition $X_0 \subseteq P_S$ is more likely to be satisfied, and, therefore, we may increase the possibility to synthesize the controller. However, selecting smaller η_x, η_u leads to a heavier computation load of synthesizing the controller, due to the evaluation of* $\mathrm{Pre}(\cdot)$ *in* (6). *Thus, users may carefully select η_x, η_u by considering the trade-off between the possibility to synthesize the controller and the computation load for the controller synthesis.* □

Remark 2. *Even though $X_0 \nsubseteq P_S$ does not hold, one might be still interested in finding a subset of initial states $X^*_0 \subseteq X_0$, such that every controlled trajectory from $x_0 \in X^*_0$ achieves reachability and safety. This subset can be derived by taking the intersection between X_0 and P_S, i.e., $X^*_0 = X_0 \cap P_S$.* □

V. ILLUSTRATIVE EXAMPLE

In this section we illustrate the effectiveness of the proposed approach through a simulation example. We consider the problem of controlling a vehicle on a road with a given road grade, which is expressed by the mapping $\alpha : [0, 1400] \to [-\pi/2, \pi/2]$, where $\alpha(x_1) = 0$, $\forall x_1 \in [0, 400)$, $\alpha(x_1) = -\frac{\pi}{120}\sin\frac{x_1 - 400}{600}\pi$, $\forall x_1 \in [400, 1000)$, $\alpha(x_1) = 0$, $\forall x_1 \in [1000, 1400]$, where x_1 denotes the position of the vehicle. The geometry of the road based on the given α is illustrated in Fig. 2. In the figure, $\gamma(\cdot)$ denotes the elevation of the road obtained by $\gamma(x_1) = \gamma(0) + \int_0^{x_1} \alpha(x)\mathrm{d}x$ with

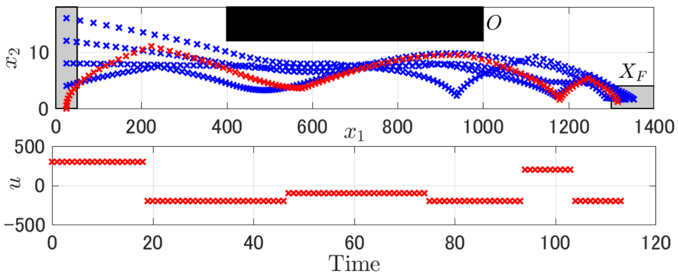

Fig. 3. Simulation results. The upper figure illustrates some trajectories by applying (8) (blue/red dotted lines). The lower figure illustrates the control inputs with $x_0 = [25; 0]$ (corresponding to the red trajectory in the upper figure).

$\gamma(0) = 10$. Let $x_k = [x_{1,k}; x_{2,k}] \in \mathbb{R}^2$ be the state, where $x_{1,k}$ is the position and $x_{2,k}$ is the velocity of the vehicle at k. We assume that the dynamics of the vehicle is given by

$$x_{1,k+1} = x_{1,k} + x_{2,k}\Delta, \tag{9}$$

$$x_{2,k+1} = x_{2,k} + \frac{1}{m_v}(u_k - h(x_{1,k}, x_{2,k}))\Delta, \tag{10}$$

where $u_k \in U$ is the force applied to the vehicle as the control input, m_v is the mass of the vehicle, and Δ is the sampling time period. $h(x_1, x_2)$ is the nonlinear term given by $h(x_1, x_2) = d_1 x_2 + d_2 m_v g \cos(\alpha(x_1)) + m_v g \sin(\alpha(x_1))$, where d_1 is the aerodynamic drag coefficient, d_2 is the rolling force coefficient, and g is the gravitational constant. The parameter settings as well as the initial, safety, target sets are illustrated in Table I.

Based on the above setting, we define the symbolic model $\widetilde{\Sigma}^{\varepsilon}_{A}$ with $\varepsilon = \eta_x = 1$, $\eta_u = 50$, $M_{\max} = 30$, and Algorithm 1 has been implemented to synthesize the controller. The execution time for the algorithm to be terminated is 20760s on Windows 10, Intel(R) Core(TM) 2.40GHz, 8GB RAM. It has been shown that $X_0 \subseteq P_S$ and thus from Theorem 1 the controller in (8) is valid for the system (1). Fig. 3 (upper) illustrates some trajectories by applying the resulting controller in (8), and Fig. 3 (lower) illustrates the applied control inputs from $x_0 = [25; 0]$. The figure shows that all trajectories enter X_F while remaining in X_S for all times, and, moreover, control inputs are updated aperiodically (with 6 number of communication times) according to the derived self-triggered strategy. For comparisons, we have also implemented Algorithm 1 with $M_{\max} = 1$ (i.e., periodic communication). The resulting trajectory from $x_0 = [25; 0]$ requires 109 number of communication time steps to achieve reachability. Hence, the self-triggered controller (Fig. 3) achieves a more communication reduction than the periodic scheme, which shows the effectiveness of the proposed approach.

VI. Conclusions and Future Work

In this paper, we proposed a symbolic approach to synthesizing a self-triggered controller with reachability and safety specifications. The symbolic model was constructed based on the notion of approximate alternating simulation relations, and a controller was synthesized for the symbolic system via a reachability game, which was then refined to a controller for the original control system. Our future work involves providing an efficient algorithm to find suitable quantization parameters η_x, η_u, such that the condition $X_0 \subseteq P_S$ in Theorem 1 is satisfied. Moreover, we would like to investigate improving the scalability of constructing symbolic models by employing compositional techniques, e.g., in [19], [20].

References

[1] W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," in *Proceedings of the 51st IEEE Conference on Decision and Control (IEEE CDC)*, 2012, pp. 3270–3285.

[2] Q. Liu, Z. Wang, X. He, and D. Zhou, "A survey of event-based strategies on control and estimation," *Systems Science & Control Engineering*, vol. 2, no. 1, pp. 90–97, 2014.

[3] C. Peng and F. Li, "A survey on recent advances in event-triggered communication and control," *Information Sciences*, vol. 457, pp. 113–125, 2018.

[4] M. A. Khatib, A. Girard, and T. Ding, "Self-triggered control for sampled-data systems using reachability analysis," in *Proceedings of 20th IFAC World Congress*, 2017, pp. 7881–7886.

[5] N. Meslem and C. Prieur, "Event-triggered algorithms for continuous-time systems based on reachability analysis," in *Proceedings of 52nd IEEE Confernece on Decision and Control*, 2013.

[6] K. Hashimoto, S. Adachi, and D. V. Dimarogonas, "Aperiodic sampled-data control via explicit transmission mapping: a set-invariance approach," *IEEE Transactions on Automatic Control*, vol. 63, no. 10, pp. 3523–3530, 2018.

[7] K. Hashimoto, S. Adachi, and D. V. Dimarogonas, "Self-triggered control for constrained systems: a contractive set-based approach," in *Proceedings of 2017 American Control Conference*, 2017, pp. 1011–1016.

[8] F. Brunner, W. Heemels, and F. Allgower, "Event-triggered and self-triggered control for linear systems based on reachable sets," *Automatica*, vol. 101, pp. 15–26, 2019.

[9] K. Hashimoto and D. V. Dimarogonas, "Synthesizing communication plans for reachability and safety specifications," *IEEE Transactions on Automatic Control*, (to appear).

[10] P. Tabuada, *Verification and Control of Hybrid Systems – A Symbolic Approach*, Springer, 2009.

[11] Z. Kader, A. Girard, and A. Saoud, "Symbolic models for incrementally stable switched systems with aperiodic time sampling," in *Proceedings of 6th IFAC Conference on Analysis and Design of Hybrid Systems*, 2018.

[12] A. S. Kolarijani and M. Mazo Jr., "A formal traffic characterization of LTI event-triggered control systems," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 1, pp. 274–283, 2018.

[13] G. Pola, A. Girard, and P. Tabuada, "Approximately bisimilar symbolic models for nonlinear control systems," *Automatica*, vol. 44, no. 10, pp. 2508–2516, 2008.

[14] A. Saoud and A. Girard, "Optimal multirate sampling in symbolic models for incrementally stable switched systems," *Automatica*, vol. 98, pp. 58–65, 2018.

[15] M. Zamani, G. Pola, M. Mazo Jr., and P. Tabuada, "Symbolic models for nonlinear control systems without stability assumptions," *IEEE Transactions on Automatic Control*, vol. 57, no. 7, pp. 1804–1809, 2012.

[16] A. Borri, G. Pola, and M. D. D. Benedetto, "Design of symbolic controllers for networked control systems," *IEEE Transactions on Automatic Control*, 2019 (to appear).

[17] M. Mazo Jr. and P. Tabuada, "Symbolic approximate time-optimal control," *Systems & Control Letters*, vol. 60, no. 4, pp. 256–263, 2011.

[18] M. Zamani, M. Mazo Jr., M. Khaled, and A. Abate, "Symbolic abstractions of networked control systems," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 4, pp. 1622–1634, 2018.

[19] A. Saoud, P. Jagtap, M. Zamani, and A. Girard, "Compositional abstraction-based synthesis for cascade discrete-time control systems," in *Proceedings of 6th IFAC Conference on Analysis and Design of Hybrid Systems*, 2018.

[20] P. J. Meyer, A. Girard, and E. Witrant, "Compositional abstraction and safety synthesis using overlapping symbolic models," *IEEE Transactions on Automatic Control*, vol. 63, no. 6, pp. 1835–1841, 2018.